



CENTRO UNIVERSITÁRIO DE BRASÍLIA- UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

ROGÉRIO ESMERALDO LEITE

CONTROLE DE CORES DE UMA LUMINÁRIA UTILIZANDO LED RGB

BRASÍLIA – DF

1º SEMESTRE DE 2013

ROGÉRIO ESMERALDO LEITE

CONTROLE DE CORES DE UMA LUMINÁRIA UTILIZANDO LED RGB

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof.^a. MSc. Maria Marony Sousa Farias

Brasília

Julho, 2013

ROGÉRIO ESMERALDO LEITE

CONTROLE DE CORES DE UMA LUMINÁRIA UTILIZANDO LED RGB

Trabalho apresentado ao Centro Universitário de Brasília (UnICEUB) como pré-requisito para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof.^a. MSc. Maria Marony Sousa Farias

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro de Computação e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS.

Prof. Abiezer Amarília Fernandes
Coordenador do Curso

Banca Examinadora:

Prof.^a MSc. Maria Marony Sousa Farias
Orientadora

Prof.^o MSc. Luís Cláudio Lopes de Araújo
Uniceub

Prof.^o Dr. Cleber da Silva Pinheiro
Uniceub

Prof.^o MSc. Fernando Chagas Santos
Uniceub

Dedico este trabalho à minha família, que me ensinou que as maiores virtudes de uma pessoa são humildade e honestidade e estão comigo em todos os momentos da minha vida, a minha futura esposa e maior incentivadora e aos meus amigos e colegas que caminharam junto a mim nessa jornada de conhecimentos e aprendizados.

Rogério Esmeraldo Leite

AGRADECIMENTOS

Agradeço sempre a Deus, por toda as bênçãos concebidas em minha vida e por iluminar e guiar meus passos.

Agradecimento eterno ao meu pai Wagner da Silva Leite, que é meu anjo da guarda e mesmo não estando mais entre nós, me ensina, me protege e rege a minha vida.

Agradecimento especial a minha mãe Maria de Lourdes Esmeraldo Silva, que com tanto carinho e amor me dá paz, me conforta em qualquer momento de dificuldade e é um exemplo não só de mãe, mas de ser humano.

Agradeço meu irmão Ricardo Wagner Esmeraldo Leite, por me motivar e não me deixar desistir de nada e estar sempre comigo sempre que preciso.

Agradeço minha futura esposa Fernanda de Paula Pimenta, por ser minha maior incentivadora para eu chegar até aqui, companheira dedicada, compartilha comigo não só os melhores mas todos os momentos da minha vida.

Agradeço a todos os meus familiares, que são exemplo de união e sempre engrandeceu minha vida com ensinamentos, afeto e momentos de descontração.

Agradeço também a todos os professores que fizeram parte dessa caminhada, sem eles eu não estaria aqui. Em particular a minha orientadora Maria Marony pela dedicação e apoio neste projeto e também durante todo o curso.

A todos os meus colegas de turma que se tornaram meus grandes amigos, em especial ao Ayrton Gissoni que me estendeu as mãos nos momentos em que mais precisei.

Deus abençoe a todos!

Rogério Esmeraldo Leite

" Só existem dois dias no ano que nada pode ser feito.
Um se chama ontem e o outro se chama amanhã,
portanto hoje é o dia certo para amar, acreditar, fazer e
principalmente viver"

Dalai Lama

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO	13
1.1 Contextualização do Projeto	13
1.2 Objetivo Geral	15
1.3 Objetivos Específicos.....	15
1.4 Motivação.....	15
1.5 Estruturas do trabalho	16
CAPÍTULO 2 REFERENCIAL TEÓRICO.....	17
2.1 A Iluminação.....	17
2.2 Propagação da Luz.....	18
2.3 A Iluminação Artificial	19
2.4 Lâmpadas LED	20
2.5 Bluetooth.....	20
CAPÍTULO 3 DESENVOLVIMENTO	21
3.1 Modelo Proposto	21
3.2 Descrição dos Materiais Utilizados	22
3.2.1 Arduino ATmega1280.....	22
3.2.2 Amarrino.....	26
3.2.3 Android.....	27
3.2.4 Led RGB	28
3.2.5 Shield Bluetooth 2.2	29
3.2.6 Interface gráfica	30
CAPÍTULO 4 DESENVOLVIMENTO PRÁTICO	31
4.1 Esquema elétrico.....	31
4.2 Programação do Aplicativo	32
4.3 Protótipo da luminária	33
4.4 Montagem.....	34
CAPÍTULO 5 TESTES E RESULTADOS	35
5.1 Teste do esquema elétrico.....	35
5.2 Teste do Aplicativo	36
5.3 Dificuldades encontradas / Soluções	39
CAPÍTULO 6 CONCLUSÕES.....	40

6.1	Conclusões.....	40
6.2	Proposta para Futuros Projetos	41
CAPÍTULO 7 REFERÊNCIAS.....		42
APÊNCIDE		44
ANEXOS		53
ANEXO – A.....		53
ANEXO – B		56

LISTA DE FIGURAS

Figura 2.1 - Faixa de longitude visível de onda	18
Figura 3.1 - Visão geral do projeto	21
Figura 3.2 - ArduinoMega ATmega1280	23
Figura 3.3 - IDE de Desenvolvimento do Arduino	24
Figura 3.4 - Gráfico de PWM - Modulação por largura de pulso	25
Figura 3.5 - Aplicativo Amarino	26
Figura 3.6 - LED RGB.....	28
Figura 3.7 - Shield Bluetooth.....	29
Figura 3.8 - Interface gráfica Android	30
Figura 4.1 - Esquema de circuito elétrico.....	31
Figura 4.2 - Protótipo da luminária	33
Figura 4.3 - Montagem do circuito.....	34
Figura 5.1 - Chaveamento da shield.....	35
Figura 5.2 - Tela aplicativo	35
Figura 5.3 - Tela Teste 1	36
Figura 5.4 - Resultado teste 1.....	36
Figura 5.5 - Tela teste 2	37
Figura 5.6 - Resultado teste 2.....	37
Figura 5.7 - Tela teste 3	38
Figura 5.8 - Resultado teste 3.....	38

LISTA DE TABELAS

Tabela 1 - Cores monocromáticas e seus comprimentos de onda e frequência	19
Tabela 2 - Especificações Arduino Mega ATmega1280.....	23
Tabela 3 - Especificações Shield Bluetooth	29
Tabela 4 - Características elétricas Shield Bluetooth	30
Tabela 5 - Custo do projeto.....	44

RESUMO

Neste trabalho foi proposto uma solução para integrar um dispositivo móvel a um equipamento de iluminação (Luminária), que serve de peça decorativa podendo customizar os mais diversos ambientes, visando gerar conforto e comodidade ao usuário. Esse sistema foi criado para controlar o equipamento através de uma interface gráfica que permite alterar intensidade e a cor da iluminação. Para desenvolver essa solução foi usado recursos de linguagem de programação Java, para criar a interface do sistema operacional Android, que fez a comunicação sem fio utilizando tecnologia Bluetooth, com um microcontrolador Arduino, responsável pelo controle da iluminação.

Palavras chaves: Iluminação, comunicação via Bluetooth, Android e Arduino.

ABSTRACT

This work proposes a solution to integrate a mobile device to a lighting fixture (luminaire), which serves as a decorative piece can customize the most diverse environments, in order to generate comfort and convenience to the user. This system was created to control the machine via a graphical interface that allows change intensity and color of illumination. To develop this solution was used resources of the Java programming language, to create the interface of the Android operating system, which made the wireless communication using Bluetooth technology with an Arduino microcontroller, which controls the lighting.

Keywords: Lighting, communication via Bluetooth, Android and Arduino.

CAPÍTULO 1 INTRODUÇÃO

1.1 Contextualização do Projeto

Desde o princípio da sua existência, o homem vem buscando um meio de sair “da escuridão”. Com o passar tempo, o homem chegou a descoberta e ao domínio de sua primeira fonte de luz: o fogo. Posteriormente, foram desenvolvidas novas fontes de luz que auxiliaram na iluminação, como: lampiões, tochas e velas.

Posteriormente com a descoberta da eletricidade foi desenvolvida a lâmpada elétrica, a qual é utilizada até os dias de hoje, as luminárias foram criadas e possibilitaram obter níveis adequados de iluminação artificial para o desenvolvimento de tarefas visuais, visando uma das principais necessidade do usuário que é o conforto e a satisfação.

A iluminação também é responsável por valorizar a qualidade do ambiente, voltado para qualquer que seja o espaço, decorando-o através da customização de cores, que é um dos principais requisitos para realçar qualquer acontecimento no ambiente, como por exemplo: a cromoterapia, entretenimento, sinalizações de trânsito e decoração de ambiente.

Com o recurso de cores de iluminação é possível auxiliar até mesmo no tratamento de doenças que é o caso da cromoterapia, que tem sido utilizada pelo homem desde as antigas civilizações, como no Egito Antigo, assim como na Índia, na Grécia, na China onde suas aplicações terapêuticas foram verificadas por experimentações e observações de resultados. (SILVA & MONTEIRO, 2006; RENE NUNES, 1990; BALZANO,2008; SITE: [HTTP:// WWW.CROMOTERAPIA.ORG.BR](http://www.cromoterapia.org.br), 2013).

A cromoterapia é a ciência que utiliza as cores do espectro solar para alterar ou manter as vibrações e o equilíbrio físico-energético em áreas do corpo humano atingidas por alguma disfunção. (SILVA & MONTEIRO, 2006; RENE NUNES, 1990; BALZANO,2008; SITE: [HTTP:// WWW.CROMOTERAPIA.ORG.BR](http://www.cromoterapia.org.br), 2013).

De acordo com a Ondina Baleano as sete cores do espectro utilizadas são:

Vermelho - ativador da corrente sanguínea;

Laranja – revitalização de ossos e músculos;

Amarelo - regenerador da pele, ossos e músculos;

Verde - antisséptico e estruturador;

Azul - analgésico, calmante e regenerador;

Anil - coagulante (em hemorragias);

Violeta - anti-infeccioso e bactericida (função de antibiótico).

Como forma de entretenimento, a utilização da iluminação, tendo em vista que o mercado de shows e eventos artísticos no Brasil, de acordo com a ICCA (Associação Internacional de Congresso e Convenções), tem crescido bastante nos últimos anos e o País chega em 7ª colocação no ranking entre os destinos mais procurados do mundo para grandes encontros de eventos. A iluminação torna-se fundamental para produção de imagens e vídeos. (SITE: [HTTP://WWW.ICCAWORLD.COM](http://WWW.ICCAWORLD.COM), 2013).

O uso da iluminação como forma de sinalização tem aplicabilidade bem prática no uso de semáforos, que podem exercer três funções: controlar o fluxo de pedestre, controlar o fluxo de veículos e controlar o fluxo de veículos e pedestres ao mesmo tempo, tendo as cores padronizadas para cada função. Ele pode ter de duas a três cores, sendo mais comum possuir três cores: a vermelha, a amarela e a verde. (SITE: [HTTP://WWW.DETRAN.MT.GOV.BR](http://WWW.DETRAN.MT.GOV.BR), 2013).

A iluminação aplicada para a decoração de ambientes, tem como papel a escolha do tom de cor que se aproxima mais de seu objetivo. Existem dois grandes grupos de cores: os quentes e os frios. As cores quentes vão do vermelho ao amarelo e as frias do azul ao verde. Resultados harmoniosos são possível utilizando as combinações entre as cores quentes com as quentes, e as frias com as frias, pois segundo a ABRA – Academia Brasileira de Arte, é comprovado que algumas combinações de cores são mais agradáveis do que se usadas de forma aleatória. (SITE: [HTTP://WWW.ABRA.COM.BR](http://WWW.ABRA.COM.BR), 2013).

1.2 Objetivo Geral

O objetivo desse projeto é projetar uma luminária que permita customizar um ambiente por meio do controle de cor e intensidade da luz, de forma remota, utilizando um dispositivo móvel que execute o sistema operacional Android.

1.3 Objetivos Específicos

Os objetivos específicos desse projeto são:

1. Desenvolvimento de um aplicativo para dispositivo móvel que execute o sistema operacional Android 2.3.3, para controlar a cor e a intensidade da luz de uma luminária.
2. Criação de um esquema elétrico com Led RGB ligado a um microcontrolador Arduino.
3. Integração do aplicativo com o esquema elétrico, por meio de comunicação Bluetooth.

1.4 Motivação

Pensando na comodidade, conforto e na customização dos mais variados ambientes, sejam: residenciais, corporativos, surgiu a motivação para criar um protótipo de luminária que pudesse atender essa mobilidade, ou seja, poder fazer uma decoração de acordo com a necessidade desejada, variando a cor e customizando-a de acordo com cada circunstância, tudo isso controlado através de uma interface gráfica para Smartphone que execute sistema operacional Android.

1.5 Estruturas do trabalho

A estrutura do trabalho se divide em capítulos:

No capítulo 2 são abordados conceitos teóricos de luminosidade, como: propriedade, velocidade, propagação e fontes de luz.

No capítulo 3 são apresentados o modelo proposto e a descrição dos componentes utilizados no projeto.

No capítulo 4 é relatada a construção do projeto.

No capítulo 5 são mostrados os testes e acompanhamentos do desenvolvimento do projeto.

No capítulo 6 são apresentadas as conclusões e sugestões de trabalhos futuros.

CAPÍTULO 2 REFERENCIAL TEÓRICO

2.1 A Iluminação

Um ponto que devemos relevar no conceito de iluminação é manter as condições adequadas para o desenvolvimento das atividades visuais, levando qualidade e conforto para o usuário.

Segundo Rocha Bicudo, um sistema de iluminação deve possuir os seguintes requisitos:

- Ter suficiência: significa que para cada ponto luminoso deve fornecer a quantidade de luz necessária a cada tipo de trabalho em virtude das variações da intensidade da luz;
- Estar sempre constante e uniformemente distribuído: de modo que evite o cansaço dos olhos. Devemos obedecer algumas condições para obtermos um ambiente confortável e com uma boa percepção visual:
 - Estimular a limpeza e a ordem (cores claras);
 - Melhorar o rendimento luminoso;
 - Evitar distribuição desigual de luz;
 - A combinação de cores;
 - Propiciar iluminação diferenciada para áreas descanso e ambientes de trabalho;
 - Utilizar iluminação com nível apropriado à função da sala.

(ROCHA BICUDO, 2009).

2.2 Propagação da Luz

Existem diferentes classificações para os materiais, ao serem atravessados por raios de luz, pelo fato de se comportarem de forma diferentes, são eles: meios transparentes, translúcidos e opacos.

O meio que permite a passagem regular da luz, sendo que é possível ver um objeto com total visibilidade por meio desse meio. Exemplos de meio transparentes são: acrílico e vidro.

Através do meio translúcido só é possível ver um objeto, porém não com total nitidez, pois a propagação da luz é irregular. Exemplo: vidro fosco.

Já no meio opaco não é possível visualizar o objeto através deste, pelo fato que não permite a passagem da luz. Exemplo: madeira.

Algumas propriedades da luz: propaga-se em todas as direções do espaço; além de ter no vácuo, velocidade igual a 300.000 km/s. A Figura 2.1 representa as cores azul, verde e vermelho, no entanto, podem ser vistas pelo ser humano as cores: violeta ou anil, azul, verde, amarela, alaranjada e vermelha, ou seja, as cores que estão compreendidas entre 400nm e 700nm de longitude de onda. (SITE: [HTTP://SOFISICA.COM.BR](http://SOFISICA.COM.BR), 2013).

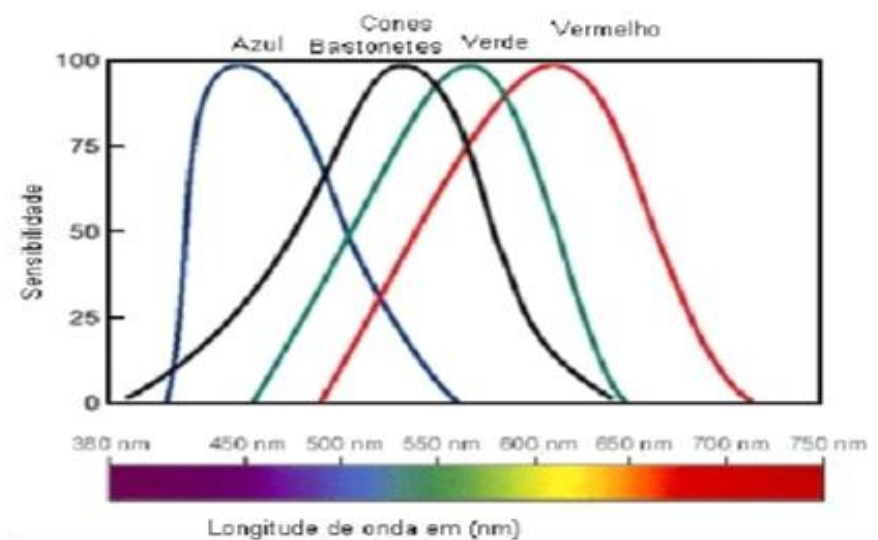


Figura 2.1 - Faixa de longitude visível de onda

Fonte: Gadelha 2007

A velocidade é inversamente proporcional à frequência pela relação entre o comprimento de onda λ e frequência f , a Tabela 1 mostra graficamente essa relação, onde o comprimento da onda é dado pela divisão da velocidade da onda (no caso a velocidade da luz ($c = 3 \times 10^8 \text{m/s}$)), pela frequência da onda.

Tabela 1 - Cores monocromáticas e seus comprimentos de onda e frequência

Fonte: Gadelha 2007



Cor	Comprimento de onda	Frequência
Vermelho	~ 625-740 nm	~ 480-405 THz
Laranja	~ 590-625 nm	~ 510-480 THz
Amarelo	~ 565-590 nm	~ 530-510 THz
Verde	~ 500-565 nm	~ 600-530 THz
Azul	~ 485-500 nm	~ 620-600 THz
Anil	~ 440-485 nm	~ 680-620 THz
Violeta	~ 380-440 nm	~ 790-680 THz

2.3 A Iluminação Artificial

A luz artificial após o surgimento da energia elétrica, tem sido uma das principais parte em trabalhos e projetos de arquiteturas visuais modernas e decoração. Além de gerar conforto para que sejam realizadas tarefas a iluminação artificial proporciona economia. A quantidade de luz, uniformidade da iluminação e ofuscamento são alguns conceitos que devem ser considerados ao criar um projeto da iluminação. Um sistema de iluminação artificial será eficiente se levar em consideração às características técnicas e o rendimento de um conjunto de elementos, dentre os quais se destacam:

- Lâmpadas;
- Luminárias;
- Circuitos de distribuição e controle;
- Decoração de ambientes;
- Cores das superfícies internas.

2.4 Lâmpadas LED

O LED ou diodo emissor de luz é um componente eletrônico semicondutor, mesma tecnologia utilizada nos chips dos computadores, tem a funcionalidade de transformar energia elétrica em luz. Essa transformação se difere da encontrada nas lâmpadas convencionais que utilizam filamentos metálicos, radiação ultravioleta e descarga de gases, dentre outras. O LED é um componente do tipo bipolar, ou seja, tem um terminal chamado anodo e outro, chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.

O LED possui um tempo de vida útil em média de 50 mil horas, se ligado durante 8 horas por dia alcança até 17 anos de uso. Comparado, por exemplo, com uma lâmpada fluorescente compacta, esse tempo chega no máximo à 10 mil horas. (OSRAM, 2013)

Maior vida útil, custos de manutenção reduzidos, eficiência, baixa voltagem de operação, controle dinâmico de cores, acionamento instantâneo, ecologicamente correto e ausência de infravermelho são alguns dos benefícios do Led. (OSRAM,2013)

2.5 Bluetooth

Bluetooth é uma tecnologia sem fio, com baixo custo, consumo de energia e tem um alcance curto porém oferece recursos de segurança avançados e hoje é encontrado em vários tipos de dispositivos, desde celulares, computadores a produtos domésticos. Atualmente o Bluetooth se encontra na versão 4.0, reforçando cada vez mais recursos de baixo consumo de energia e velocidade aprimorada, que serve de base para dispositivos inteligentes, ele permite uma alta performance para enviar e receber vídeos, músicas entre outros arquivos, de forma simples entre os dispositivos. (SITE: [HTTP://WWW.BLUETOOTH.COM](http://www.bluetooth.com), 2013).

CAPÍTULO 3 DESENVOLVIMENTO

3.1 Modelo Proposto

O modelo proposto é um protótipo de uma luminária personalizada com o objetivo de gerar conforto e satisfação ao usuário para customizar os mais variados tipos de ambientes. As propriedades de cores e intensidade da luminária são controladas por meio de um aplicativo para dispositivo móvel executando sistema operacional Android, utilizando a conexão via Bluetooth.

Conforme mostrado na Figura 3.1, no primeiro passo o dispositivo móvel com Android, envia o comando para a Shield Bluetooth por meio de comunicação Bluetooth. No segundo passo a Shield Bluetooth recebe o comando e envia para o Arduino, que no terceiro passo, recebe o comando e envia os dados para o Led RGB.

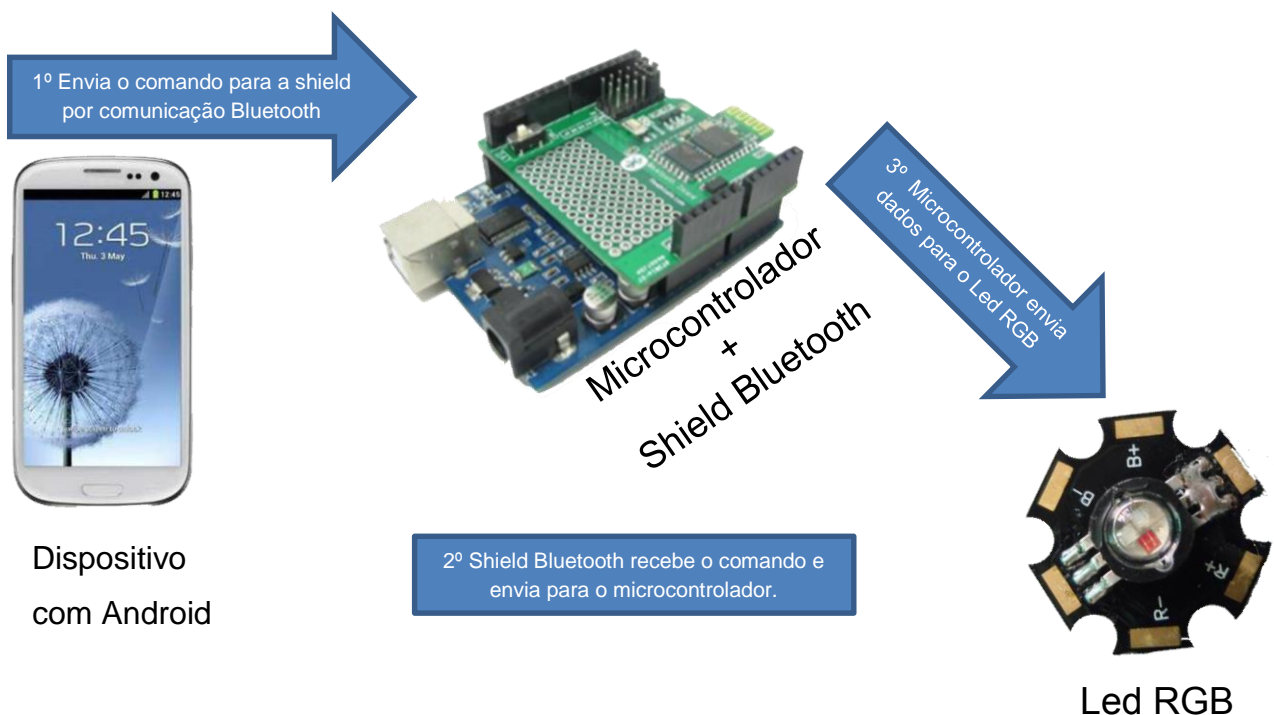


Figura 3.1 - Visão geral do projeto

Autor: Rogério Leite

3.2 Descrição dos Materiais Utilizados

3.2.1 Arduino ATmega1280

No projeto, foi utilizado o microcontrolador Arduino ATmega1280 como é mostrado na Figura 3.2, que é uma plataforma de computação física que possui um microcontrolador de placa única com suporte embutido de entrada e saída.

O Arduino é uma plataforma de desenvolvimento que possui software e hardware de distribuição livre, onde qualquer pessoa interessada poderá se basear nos códigos e nas placas, podendo assim fazer a sua própria placa. Uma característica do Arduino é ser multi-plataforma, ou seja, roda em Windows, Mac e Linux.

Utilizando o Arduino, é possível fazer diversas aplicações interagindo com o ambiente através da criação de programas que manipulem suas entradas e saídas, convertendo medidas do mundo real em sinais elétricos.

Utilizando a entrada e saída do Arduino, é possível conectar diversos sensores como o sensor de temperatura, luminosidade, umidade, pressão, distância e outros dispositivos como o displays, LED, interruptores, motores e placas extensíveis como GSM, 3G, Ethernet, Bluetooth, relés e etc.

O Arduino Mega possui 54 entradas/saídas digitais, das quais 14 podem ser utilizadas como saídas PWM, 16 entradas analógicas, 4 portas seriais (UART). O Arduino Mega utiliza um oscilador de 16MHz, possui conector de força, conector ICSP, botão de reset e uma interface de comunicação USB. (SITE: [HTTP://WWW.ARDUINO.CC](http://www.arduino.cc), 2013).

Na Tabela 2 é mostrada a especificação do Arduino ATmega 1280, é possível ver todas as especificações técnicas no ANEXO-B.

Tabela 2 - Especificações Arduino Mega ATmega1280

Microcontrolador	ATmega1280
Tensão de Alimentação	5V
Tensão de Entrada (recomendado)	5-12V
Tensão de Entrada (limites)	6-20V
Portas Digitais I/O	54
Portas Analógicas de Entrada	16
Corrente DC por pino	40mA
Corrente DC por pino 3.3V	50mA
Memória Flash	128KB
SRAM	8KB
EEPROM	4KB
Clock	16MHz

Fonte: www.Arduino.cc

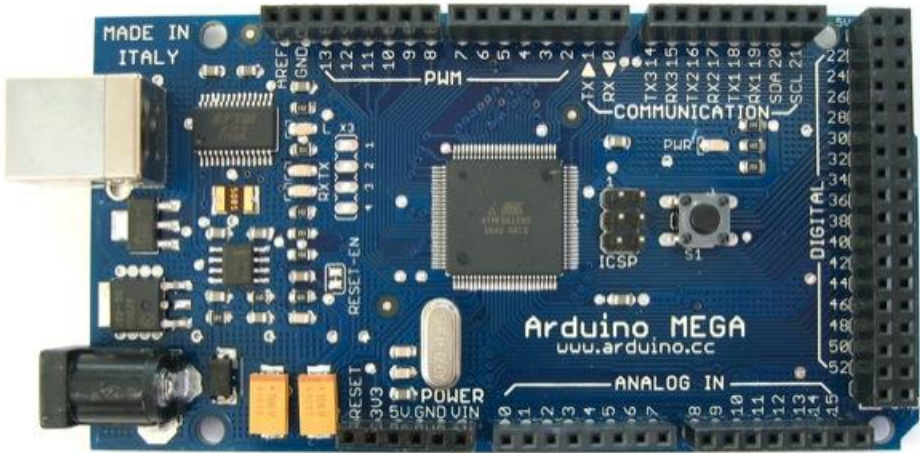


Figura 3.2 - ArduinoMega ATmega1280

Fonte: www.arduino.cc

Para a programação no Arduino, é necessário utilizar a IDE própria desenvolvida em software aberto, por meio da IDE podemos programar, fazer a compilação do código e com o código já compilado é possível fazer o upload e gravá-lo na placa.

O ambiente de desenvolvimento IDE do Arduino, utiliza a linguagem de programação *Wiring*, que é um tipo de linguagem variante do C e do C++. O ambiente de desenvolvimento já possui algumas bibliotecas específicas, é possível também adicionar outras bibliotecas ou até mesmo criá-las. Na Figura 3.3 é mostrado o ambiente de desenvolvimento para o Arduino.

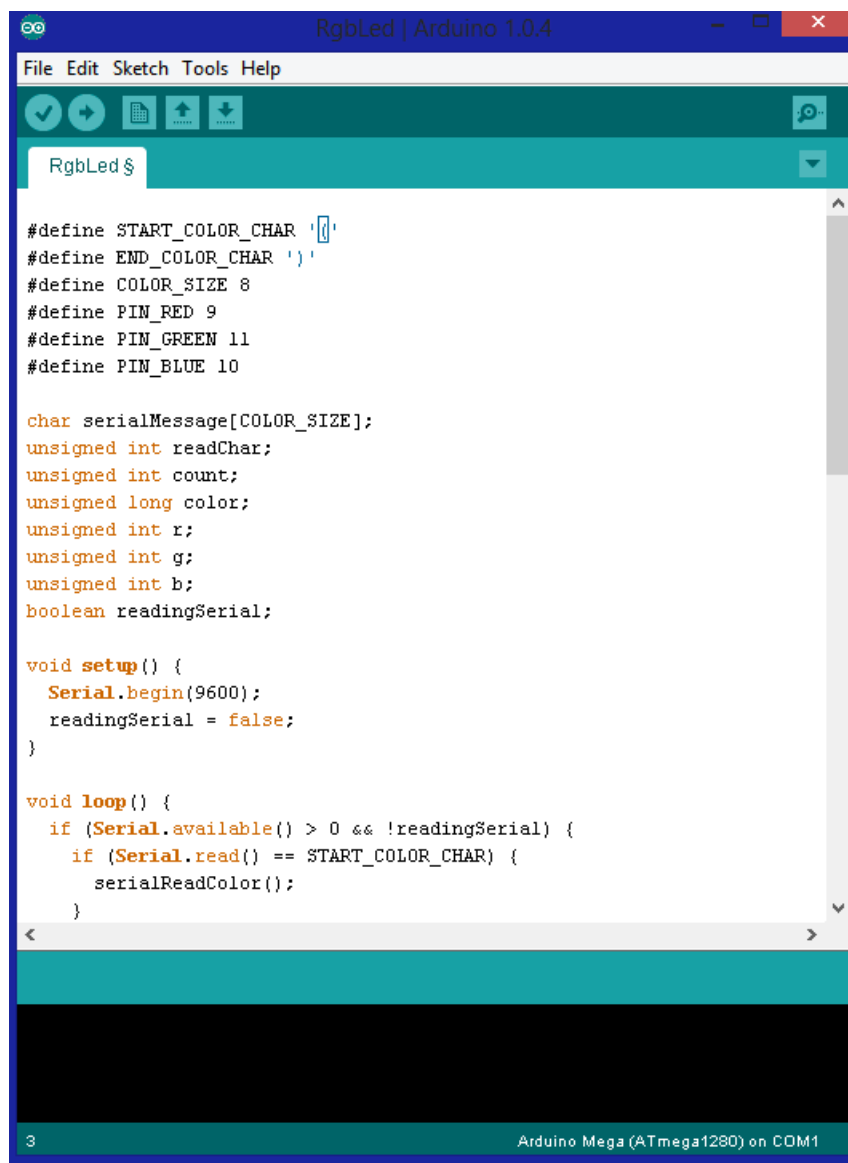


Figura 3.3 - IDE de Desenvolvimento do Arduino

Autor: Rogério Leite

A escolha do Arduino se deve pela facilidade com que ele pode ser conectado com outros dispositivos e sensores, por ele ser multi-plataforma, por possuir uma linguagem simples e acessível, por ser de fácil aprendizado, possui uma quantidade enorme de material de estudo, indo de livros, vídeo aulas a fóruns e por ser uma placa de alta disponibilidade no mercado.

O Arduino ATmega1280 possui as portas *PWM*, identificadas por “PWM” impresso na placa do microcontrolador e que significa, *Pulse Width Modulation* ou modulação por largura de pulso (modulação que fornece um sinal analógico por meios digitais). Essa forma de onda do sinal digital é demonstrada em uma onda quadrada que alterna seu estado em nível lógico alto e nível lógico baixo, pode ser representado pelo sistema binário 0 e 1.

Como é mostrado na Figura 3.4, *Duty Cycle* é a razão entre o período de pico e o período total da onda, então para ter uma onda quadrada real, isso significa que possuem picos e vales iguais, é preciso que o *Duty Cycle* seja exatamente 50%, sendo 50% de pico e os outros 50% de vale. (SITE: [HTTP://WWW.ARDUINO.CC](http://www.arduino.cc), 2013).

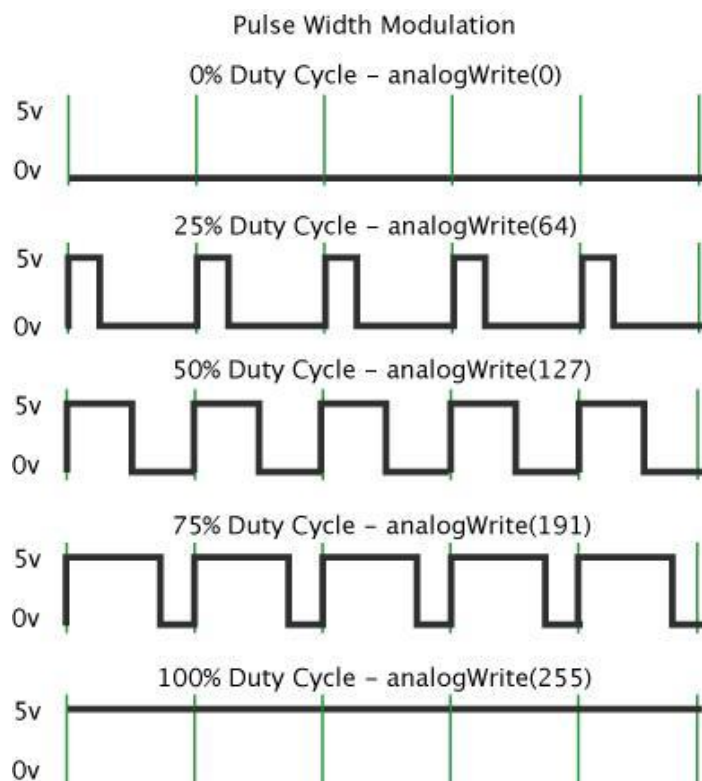


Figura 3.4 - Gráfico de PWM - Modulação por largura de pulso

Fonte: <http://Arduino.cc/en/Tutorial/PWM>

3.2.2 Amarino

O Amarino começou como um projeto no *MIT Media Lab*, ele é um kit de ferramentas que serve para conectar qualquer dispositivo móvel, que execute sistema operacional Android, em microcontroladores Arduino por meio da comunicação Bluetooth.

A plataforma é open-source e oferece padrões para facilitar a interação entre o dispositivo móvel com Android e um microcontrolador Arduino, usando de recursos como: acelerômetro, sensores, leds entre outros. O Amarino possui uma API interface de programação de aplicativos, que possibilita total controle sobre a biblioteca.

A utilização do kit Amarino se dá através da importação da biblioteca *AmarinoLibrary.jar* para o projeto Android, que deve ser adicionada no ADT Android Developer Tools e para que o Arduino se comunique com a plataforma Amarino, deve ser adicionada a biblioteca *MeetAndroid* na IDE Arduino e por último a instalação do aplicativo Amarino em um dispositivo com sistema operacional Android, como é mostrada a interface na Figura 3.5.(SITE: [HTTP://WWW.AMARINO-TOOLKIT.NET](http://www.amarino-toolkit.net), 2013).



Figura 3.5 - Aplicativo Amarino

Autor: Rogério Leite

3.2.3 Android

O Android é o sistema operacional mais conhecido do mundo, ele tem sua base no núcleo Kernel Linux modificado, foi desenvolvido sob a liderança da empresa Google em conjunto com outras empresas. (SITE: [HTTP://WWW.ANDROID.COM](http://WWW.ANDROID.COM), 2013).

Foi utilizado o Android pelo fato de ser uma plataforma versátil, aberta e que suporta programas desenvolvidos na linguagem de programação Java, tem uma grande quantidade de ferramentas e APIs que auxiliam no desenvolvimento para aplicativos que rodem na plataforma Android, além de ser o mais popular sistema operacional para dispositivo móvel. (SITE: [HTTP://WWW.ANDROID.COM](http://WWW.ANDROID.COM), 2013).

Atualmente o Android está na sua versão 4.2.2 conhecida como Jelly Bean, porém foi utilizado no projeto o Android na versão 2.3.3, por ser a versão compatível com o smartphone utilizado.

3.2.4 Led RGB

O padrão RGB é uma sigla originária do inglês e significa, red (vermelho), green(verde) e blue(azul), que são as três cores primárias e é com elas que podemos criar todas as outras tonalidades possíveis. O Led RGB possui, na verdade, cada uma das três cores separadamente, sendo controladas da mesma forma, tornando esse tipo de componente útil para aplicações mais práticas.

O Led RGB funciona basicamente com mistura de cores, variando a luminosidade de uma cor e aumentando a das outras, dando origem a novas tonalidades, possui dois polos sendo um negativo e um positivo como é mostrado na Figura 3.6.

No microcontrolador Arduino pode-se controlar tal componente, para isso precisaremos de três portas PWM e o código de programação. A partição é de 8 bits para cada uma das cores (vermelho, verde e azul), sendo assim possível um alcance de 256 valores, ou intensidades, para cada uma das cores primárias. Com este sistema, mais de 16 milhões ($16.777.216$ ou 256^3) diferentes combinações de tons, saturação e brilho podem ser reproduzidos, mesmo que não possamos distingui-las com facilidade, todas as informações e descrições técnicas está no ANEXO-A.

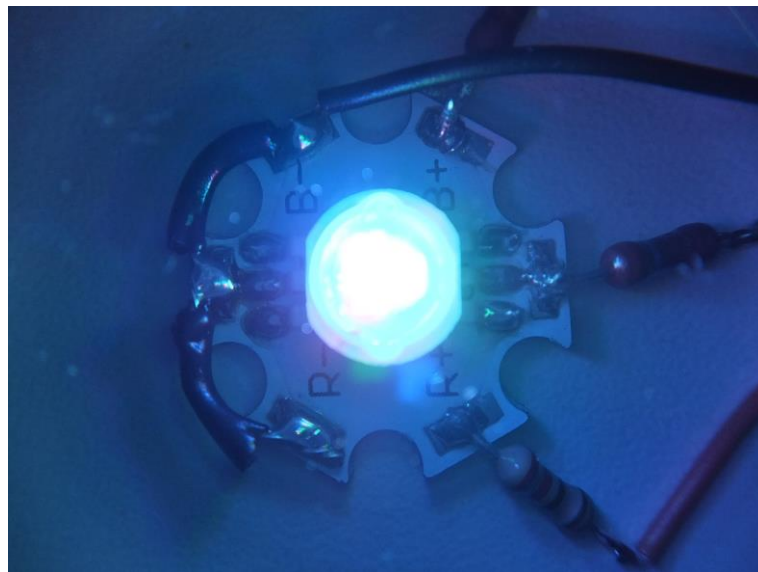


Figura 3.6 - LED RGB

Autor: Rogério Leite

3.2.5 Shield Bluetooth 2.2



Figura 3.7 - Shield Bluetooth

Fonte: <http://blog.iteadstudio.com>

BT V2.2 é um módulo Bluetooth porta serial (com o modo de mestre e escravo), como mostrado na Figura 3.7, e é compatível para uso em modo DAT e pode ser usado diretamente com Arduino,

Especificações:

Tabela 3 - Especificações Shield Bluetooth

Fonte: <http://blog.iteadstudio.com>

Microprocessor	CSR BC417
PCB size	53.3mm X 47mm X 1.6mm
Indicators	PWR State
Power supply	5V DC
IO	6
Communication Protocol	UART/Bluetooth 2.0
RoHS	Yes

Características elétricas:

Tabela 4 - Características elétricas Shield Bluetooth

Fonte: <http://blog.iteadstudio.com>

Specification		Min	Type	Max	Unit
Power Voltage		4.5	5	5.5	VDC
Input	Target Voltage = 3.3 V	3	3.3	3.6	V
Voltage	Target Voltage = 5 V	4.5	5	5.5	
VH					
Input Voltage VL:		-0.3	0	0.5	V
CurrentConsumption		-	20	40	mA

3.2.6 Interface gráfica

A interface gráfica criada para o Android, para controlar as cores do Led, é visualizada na forma de barras, um para cor vermelha, outro para cor verde e um para a cor azul, como é mostrado na Figura 3.8.



Figura 3.8 - Interface gráfica Android

Autor: Rogério Leite

CAPÍTULO 4 DESENVOLVIMENTO PRÁTICO

4.1 Esquema elétrico

Na Figura 4. é mostrado um diagrama do circuito elétrico do protótipo na protoboard, utilizando o programa *Proteus*.

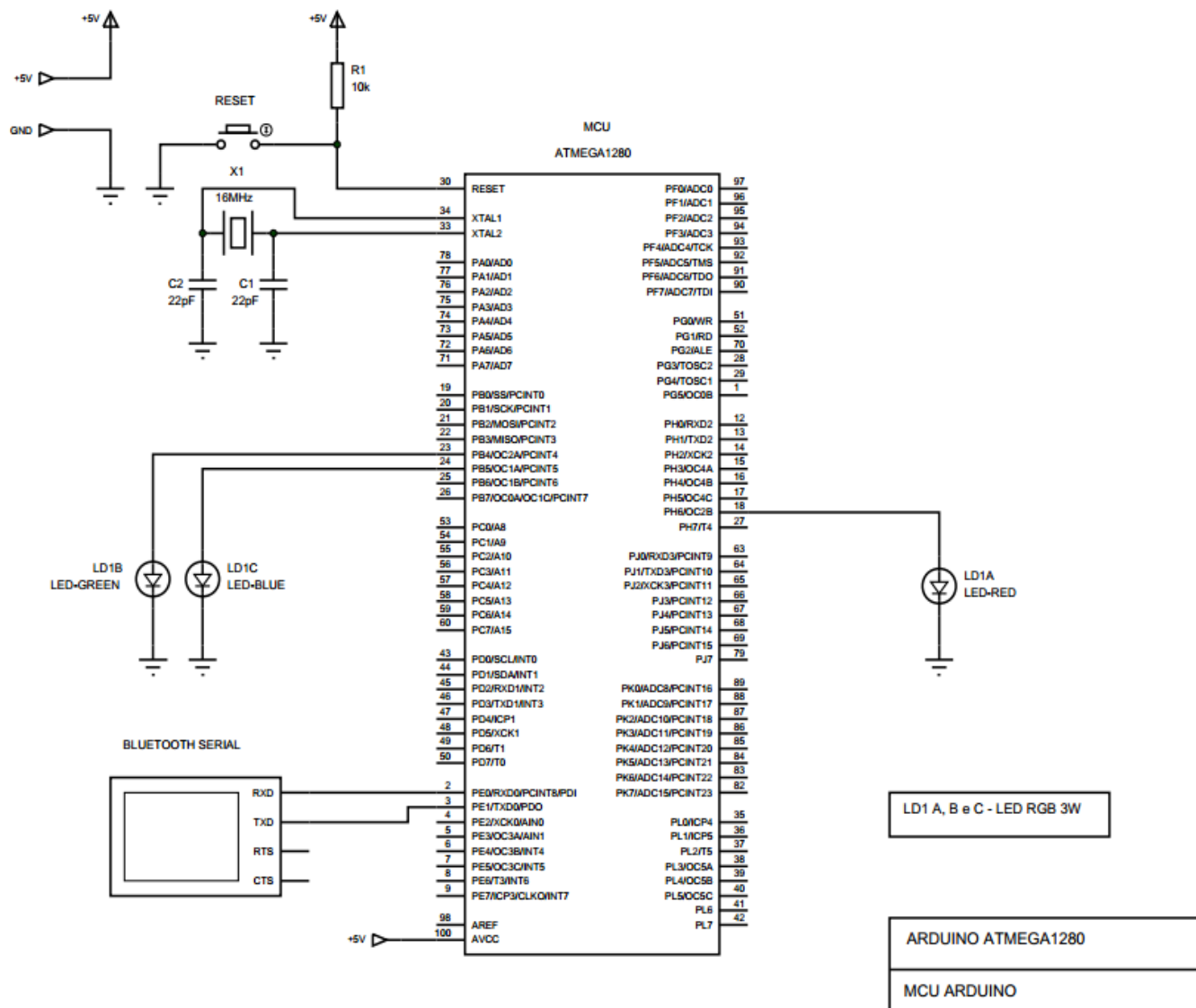


Figura 4.1 - Esquema de circuito elétrico

Autor: Rogério Leite

4.2 Programação do Aplicativo

Foi utilizado o ADT – Android Developer Tools, para a criação do aplicativo para Android, que interpreta linguagem de programação Java, compila e gera o aplicativo com a extensão, “.apk” (abreviação que vem de AndroidPackage), que é uma forma de um arquivo compactado que pode ser executado no sistema operacional Android.

A biblioteca Amarino foi utilizada para fazer a comunicação Bluetooth do Aplicativo com o microcontrolador Arduino. O aplicativo funciona com utilização de barras para controlar cada cor primária, o código fonte do desenvolvimento do controle de cores está no apêndice.

4.3 Protótipo da luminária

Foi utilizado acrílico para a montagem da luminária por ser um material versátil, com facilidade para a personalização e por poder ser usado nos mais variados tipos de ambientes, além de ser um material com alto índice de transparência, sendo excelente para que a luz se propague, oferece também segurança quanto a quebra do material, que não se estilhaça e apresenta um peso leve e ideal para se usar em ambientes internos.

O protótipo criado é para uso corporativo. O formato utilizado é correspondente a logomarca da empresa “Criattus Websites e Sistemas Online”, como é mostrado na Figura 4.2, o qual será usado para fins de decoração em seu ambiente de trabalho.

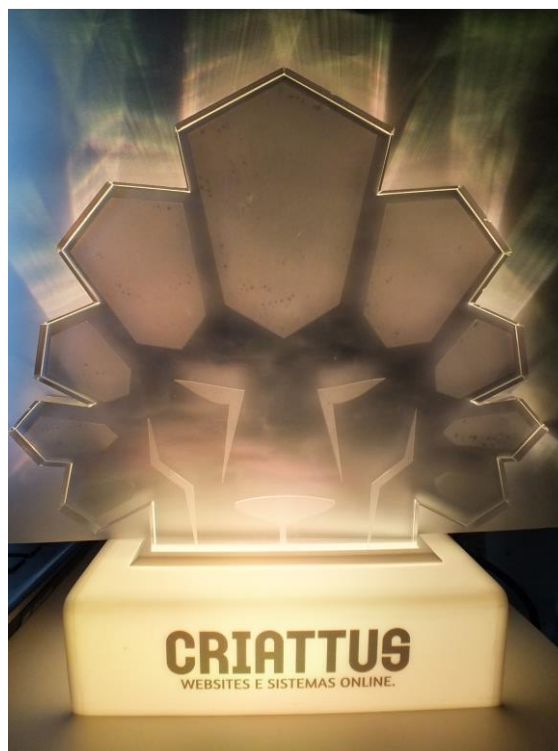


Figura 4.2 - Protótipo da luminária

Autor: Rogério Leite

4.4 Montagem

Na Figura 4.3 é mostrado o microcontrolador Arduino acoplado a uma Shield Bluetooth e ligado ao Led RGB através das portas PWM, sendo a porta 9 para a cor vermelha, porta 10 para a cor verde, porta 11 para a cor azul e o pino GND nos polos negativos do Led, referente ao aterramento, em funcionamento pela ligação da porta USB e com a chave no modo CMD (modo de comando).

Com a chave no modo CMD, foi aberto o Arduino e através do módulo serial monitor, foi configurado o nome do dispositivo através do comando, AT+NAME = ROGERIO, foi retornado ok no console, sinalizando que foi a configuração foi realizada com sucesso.

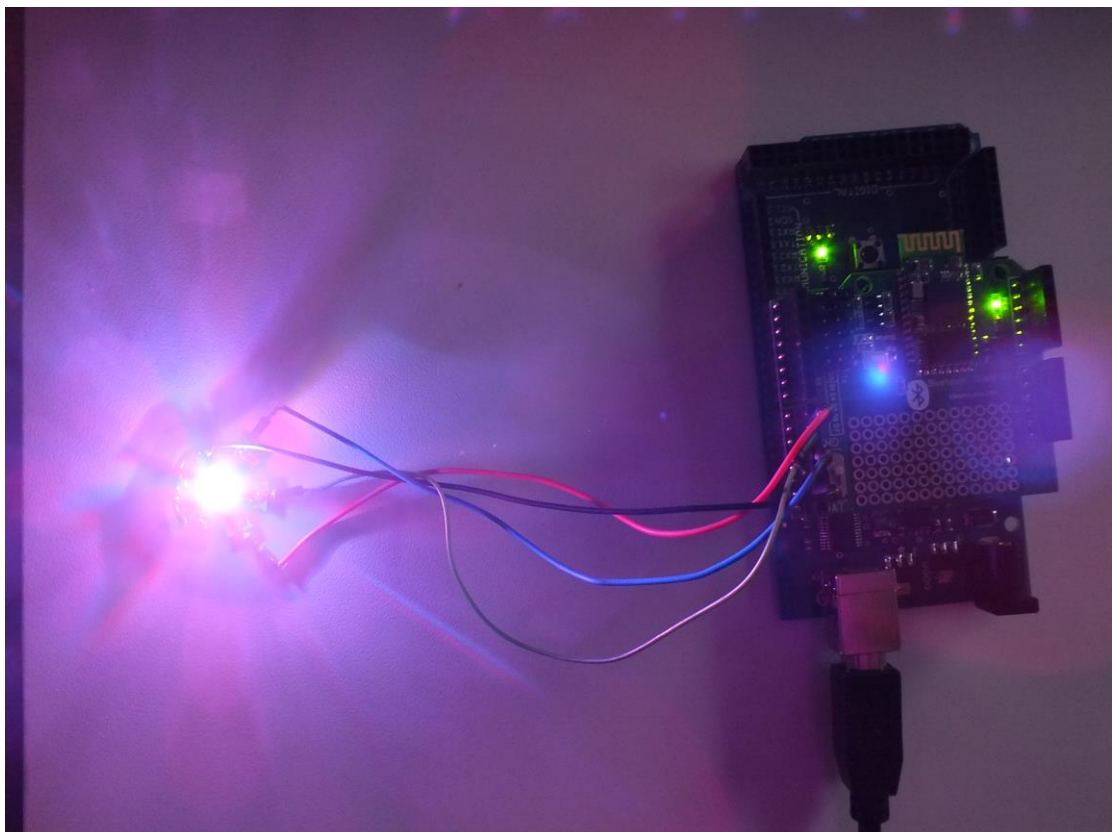


Figura 4.3 - Montagem do circuito

Autor: Rogério Leite

CAPÍTULO 5 TESTES E RESULTADOS

5.1 Teste do esquema elétrico

Para o pareamento do circuito, foi colocado a chave da Shield Bluetooth que é mostrado na Figura 5.1, na posição DAT(dados), o Led da Shield ficou piscando intermitentemente. No dispositivo com Android foi aberto o aplicativo Amarino como é mostrado na Figura 5.2, ele gera um número ID para sincronizar no dispositivo e para ativar a conexão Bluetooth, após o reconhecimento do dispositivo configurado com nome 'ROGERIO', executei a conexão e o Led parou de piscar, indicando que o Android estava sincronizado com o microcontrolador.



Figura 5.1 - Chaveamento da shield

Fonte: <http://blog.iteadstudio.com>

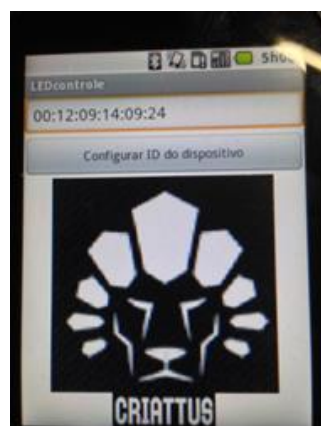


Figura 5.2 - Tela aplicativo

Autor: Rogério Leite

5.2 Teste do Aplicativo

Através do terminal de comandos da aplicação Android, após realizar a conexão Bluetooth utilizando a biblioteca Amarino, foram alteradas as barras de forma aleatória, os quais correspondem as cores de acordo com os resultados obtidos.

Para o primeiro teste, foram alteradas as barras para as posições como é mostrado na Figura 5.3, o protótipo apresentou o resultado como é mostrado na Figura 5.4.

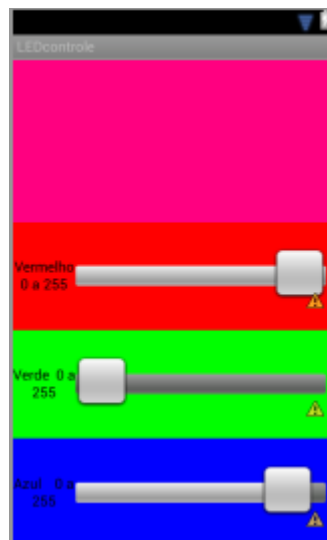


Figura 5.3 - Tela Teste 1

Autor: Rogério Leite



Figura 5.4 - Resultado teste 1

Autor: Rogério Leite

Para o segundo teste, foram alteradas as barras para as posições como mostra a Figura 5.5, o protótipo apresentou o resultado como é mostrado na Figura 5.6.

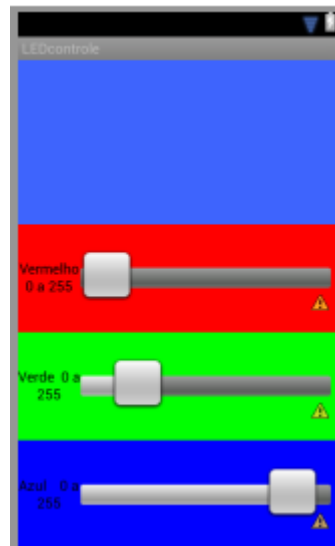


Figura 5.5 - Tela teste 2

Autor: Rogério Leite



Figura 5.6 - Resultado teste 2

Autor: Rogério Leite

Para o terceiro e último teste, foram alteradas as barras para a posição como mostra a Figura 5.7, o protótipo apresentou o resultado como é mostrado na Figura 5.8.



Figura 5.7 - Tela teste 3

Autor: Rogério Leite



Figura 5.8 - Resultado teste 3

Autor: Rogério Leite

5.3 Dificuldades encontradas / Soluções

Incompatibilidade da Shield Bluetooth com o Arduino ATmega1280, para solucionar esse problema e configurar a Shield Bluetooth, foi necessário a utilização o Arduino duemilanove uma versão passada do microcontrolador Arduino, o qual foi possível retirar o chip do microcontrolador para efetuar a configuração da Shield Bluetooth.

Outra dificuldade foi na configuração do ambiente de desenvolvimento para Android, devido a incompatibilidade de versões, para corrigir o problema a utilização de versões idênticas tanto para o dispositivo Android como para a versão do ADT foi devidamente ajustada.

CAPÍTULO 6 CONCLUSÕES

6.1 Conclusões

Com este trabalho verificou-se que o objetivo geral do projeto foi atingido, pois os resultados obtidos foram de acordo com o esperado e o controle de cores da luminária, por meio de um dispositivo com sistema operacional Android, utilizando comunicação Bluetooth teve o funcionamento devidamente contemplado.

O microcontrolador Arduino atendeu devidamente o que se esperava. Através dele, foi possível controlar as portas PWM, que enviaram os pulsos de forma correta alterando as cores do Led RGB.

A aplicação criada para controlar as cores do Led, mostrou se eficaz pois é possível que o usuário faça combinações e selecione a cor desejada de forma simples e com bons recursos visuais.

A Shield Bluetooth apesar de apresentar um problema de incompatibilidade com o microcontrolador Arduino ATmega1280, teve o problema solucionado e a Shield realizou a comunicação via Bluetooth que se esperava, recebendo e enviando os comandos da forma devida, o que possibilitou a comunicação de forma remota entre o dispositivo móvel e a luminária.

O projeto se mostra um sistema eficaz no controle de cores através de um dispositivo móvel que execute o sistema operacional Android, além de apresentar recursos de operação muito prático tendo seus recursos de funcionalidade devidamente atendidos.

6.2 Proposta para Futuros Projetos

Para projetos futuros, foi sugerido a implementação de programação de timer para ligar e desligar o Led, adicionar um display de cores *Touch Screen* diretamente na luminária para customização independente de dispositivos remotos.

Utilização de sensores de presença e calor, para automatizar a luminária, permitindo a alteração de cor de acordo com o som ambiente e até com a agitação da pessoa, aplicando mudanças conforme cada circunstância.

CAPÍTULO 7 REFERÊNCIAS

AMARINO. Disponível em: <<http://www.amarino-toolkit.net>>. Acessado em junho de 2013.

ANDROID. Disponível em: <<http://www.android.com>>. Acessado em junho de 2013.

ARDUINO. Disponível em < Fonte: www.Arduino.cc/>. Acesso em: maio de 2013.

ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DA ILUMINAÇÃO. Disponível em: <www.abilux.com.br>. Acesso em: fevereiro de 2013.

BALZANO, O. Cromoterapia: Medicina Quântica. São Paulo, 2008.

BICUDO, Glauber Rocha. Controle de iluminação utilizando controlador lógico programável. Monografia, Universidade Federal de Ouro Preto, Ouro Preto, 2009.

BLUETOOTH. Disponível em: <<http://www.bluetooth.com>>. Acessado em junho de 2013.

ABRA. **Combinação de cores**. Disponível em: <<http://www.abra.com.br/oficinas/11-como-combinar-as-cores-introducao>>. Acessado em maio de 2013.

DELIO PEREIRA GUERRINI. “Iluminação: Teoria e Projeto”. Editora Érica, 2008.

DETRAN MT. Disponível em.

<<http://www.detran.mt.gov.br/adm/uploads/downloads/72a5aanexo2.pdf>>. Acessado em março de 2013.

ICCA. Country and City ranking. Disponível em:

<http://www.iccaworld.com/dcps/doc.cfm?docid=1364 Stats_CCRankings2011.pdf>. Acessado em junho de 2013.

NUNES, Rene. Cromoterapia a cura através da cor: Esoterismo.

OSRAM. **Lâmpadas LEDS**. 2013. Disponível em:

<http://www.osram.com.br/osram_br/Consumidor/Iluminacao_Geral/LampLEDs>. Acessado em janeiro de 2013.

OSRAM. **Manual Luminotécnico Prático**. 2013. Disponível em:

<<http://www.osram.com.br/download/manual/MANUAL.PDF>>. Acessado em janeiro de 2013.

PHILIPS. Manual de Iluminação. 3. ed. Holanda: Philips Lighting Division, 1981.

PILOTTO NETO, E. Cor e Iluminação nos Ambientes de Trabalho. São Paulo: Livraria Ciência e Tecnologia Editora, 1980.

SILVA & MONTEIRO, Cromoterapia, 2006.

SILVA & MONTEIRO. Disponível em: <<http://www.cromoterapia.org.br>>. Acesso em fevereiro de 2013.

TIPLER, Paul. ***Física para Cientistas e Engenheiros***: Mecânica, Oscilações, Ondas e Termodinâmica - volume I. 5 ed. LOCAL: LTC, 2009.

UTILIUZ. Disponível em <Fonte: www.utiluz.com/>. Acessado em: maio de 2013.

Vinícius de Araújo Moreira, ***Iluminação e Fotometria*** – teoria e aplicação, Edgard Blucher Ltda, 1987.

APÊNCIDE

Visando o custo benefício para viabilidade comercial, o projeto se torna viável, sendo os custos relativos ao projeto descritos na Tabela 5, e ainda com a possibilidade de redução de custos para compras em grandes quantidades dos materiais utilizados, e ainda podendo optar por versões passadas do microcontrolador, que possuem portas PWM.

Tabela 5 - Custo do projeto

Fonte: Autor

Descrição	Valor em R\$
Microcontrolador Arduino ATmega1280	120,00 (cento e vinte)
Shield Bluetooth V2.2	50,00 (cinquenta)
Led RGB	8,00 (oito)
Protótipo luminária em acrílico	30,00 (trinta)
Total	208,00 (duzentos e oito)

Código do Arduino:

```
#include <MeetAndroid.h>

//declarando a biblioteca para chamar as funções

//selecionar os pinos PWM para controlar o led

int redLed = 9;
int greenLed = 10;
int blueLed = 11;

void setup()
{
    //usando a taxa de transmissão padrão do módulo Bluetooth
    Serial.begin(9600);

    // funções de retorno de chamada, que será chamado quando tiver um evento
    associado.

    meetAndroid.registerFunction(red, 'r');
    meetAndroid.registerFunction(green, 'g');
    meetAndroid.registerFunction(blue, 'b');

    // define os leds como pinos de saída
    pinMode(redLed, OUTPUT);
    pinMode(greenLed, OUTPUT);
    pinMode(blueLed, OUTPUT);
```

```
//definir os leds como ligados
digitalWrite(redLed, HIGH);
digitalWrite(greenLed, HIGH);
digitalWrite(blueLed, HIGH);
}

void loop()
{
    meetAndroid.receive();
}

//loop para receber eventos

//função chamada com o controle da barra vermelho no aplicativo
void red(byte flag, byte numOfValues)
{
    analogWrite(redLed, meetAndroid.getInt());
}

//função chamada com o controle da da barra verde no aplicativo
void green(byte flag, byte numOfValues)
{
    analogWrite(greenLed, meetAndroid.getInt());
}

//função chamada com o controle da barra azul no aplicativo
void blue(byte flag, byte numOfValues)
{
    analogWrite(blueLed, meetAndroid.getInt());
}
```

Código do Aplicativo Android:

```
//pacote amarino
package br.com.rogerio.luminariargb;

import android.app.Activity;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.View;
import android.widget.SeekBar;
import android.widget.SeekBar.OnSeekBarChangeListener;
import at.abraxas.amarino.Amarino;

public class Luminariargb extends Activity implements
    OnSeekBarChangeListener {

    private static final String TAG = "controleLED";

    final int DELAY = 150;

    SeekBar redSB;
    SeekBar greenSB;
    SeekBar blueSB;
    View colorIndicator;
```

```
int red, green, blue;

long lastChange;

//chamado quando a atividade é criada pela primeira vez
@Override

public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);


    // Amarino.connect(this, DEVICE_ADDRESS);


    //recupera referências às views definidas no arquivo de layout main.xml

    redSB = (SeekBar) findViewById(R.id.SeekBarRed);
    greenSB = (SeekBar) findViewById(R.id.SeekBarGreen);
    blueSB = (SeekBar) findViewById(R.id.SeekBarBlue);
    colorIndicator = findViewById(R.id.ColorIndicator);


    //registrando os ouvintes

    redSB.setOnSeekBarChangeListener(this);
    greenSB.setOnSeekBarChangeListener(this);
    blueSB.setOnSeekBarChangeListener(this);
}


@Override

protected void onStart() {

    super.onStart();
```



```

//carrega o último estado
SharedPreferences prefs = PreferenceManager
    .getDefaultSharedPreferences(this);

red = prefs.getInt("red", 0);
green = prefs.getInt("green", 0);
blue = prefs.getInt("blue", 0);


//configura as barras e a cor de fundo de acordo com o último estado
redSB.setProgress(red);
greenSB.setProgress(green);
blueSB.setProgress(blue);
colorIndicator.setBackgroundColor(Color.rgb(red, green, blue));

new Thread() {
    public void run() {
        try {
            Thread.sleep(6000);
        } catch (InterruptedException e) {
        }

        Log.d(TAG, "atualizar cores");
        updateAllColors();
    }
}.start();
}

```

```

@Override

protected void onStop() {

    super.onStop();

    //parar o serviço de tela de fundo do Amarino, não é mais necessário

    Amarino.disconnect(this,
LuminariargbComInterface.DEVICE_ADDRESS);

}

```

```

@Override

public void onProgressChanged(SeekBar seekBar, int progress,

    boolean fromUser) {

    //não enviar muitas atualizações

    if (System.currentTimeMillis() - lastChange > DELAY) {

        updateState(seekBar);

        lastChange = System.currentTimeMillis();

    }

}

```

```

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

    lastChange = System.currentTimeMillis();

}

```

```
@Override

public void onStopTrackingTouch(SeekBar seekBar) {

    updateState(seekBar);

}

private void updateState(final SeekBar seekBar) {

    switch (seekBar.getId()) {

        case R.id.SeekBarRed:

            red = seekBar.getProgress();

            updateRed();

            break;

        case R.id.SeekBarGreen:

            green = seekBar.getProgress();

            updateGreen();

            break;

        case R.id.SeekBarBlue:

            blue = seekBar.getProgress();

            updateBlue();

            break;

    }

    //fornece retorno dos utilizadores mudando a cor de fundo

    colorIndicator.setBackgroundColor(Color.rgb(red, green, blue));

}

private void updateAllColors() {

    updateRed();
```

```
        updateGreen();  
        updateBlue();  
    }  
  
    private void updateRed() {  
  
        Amarino.sendDataToArduino(this,  
                                   LuminariargbComInterface.DEVICE_ADDRESS, 'r', red);  
    }  
  
    private void updateGreen() {  
        Amarino.sendDataToArduino(this,  
                                   LuminariargbComInterface.DEVICE_ADDRESS, 'g',  
green);  
    }  
  
    private void updateBlue() {  
        Amarino.sendDataToArduino(this,  
                                   LuminariargbComInterface.DEVICE_ADDRESS, 'b', blue);  
    }  
  
}
```

ANEXOS

ANEXO – A

High Power 3W RGB LED LED-RGB-Star43CD

<https://solarbotics.com/product/LED-RGB-Star43CD/>



This 3W RGB LED can produce a brightness of 43000 millicandela, that's around 3 times brighter than the most luminous 5mm LEDs out on the current market!

WARNING! These LEDs are BRIGHT, and when we say BRIGHT, we mean that it isn't very bright to look at these BRIGHT LEDs! (or if you do, use some shades).

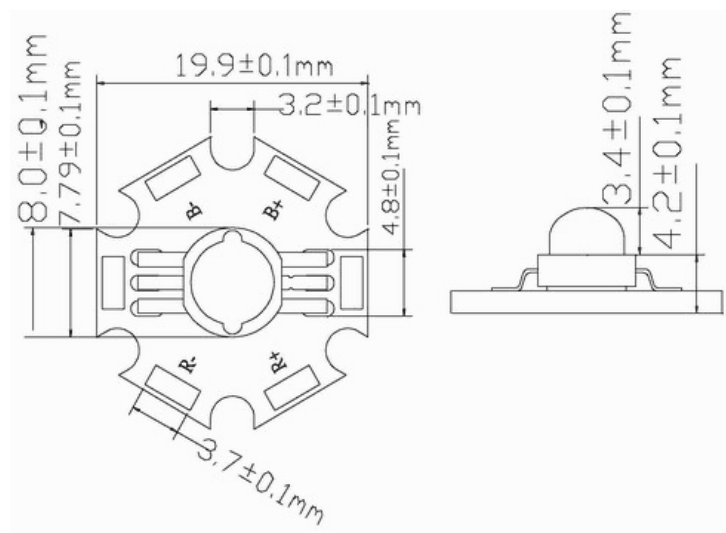
The sun is bright and these LEDs seem almost brighter.....almost. The standard rating for each LED is 350mA so try to limit the current to this value when placing these in your application. The aluminum backing will dissipate enough heat at 350mA but if you'd like to go any higher please note that you will require additional heatsinking.

These are perfect for edge-lighting & back-lighting signs and displays. Also good for mood lights or night lights, combine it with our diffused enclosure for a great party light or a rainbow patio display. Give them a shot on your robot, we've never seen robot eyes that bright before!

Now also available as part of our Star Controller RGB LED easy-interface kit

Specifications (most specs in R/G/B format):

Color (R/G/B)	Red Green Blue	Continuous Forward Current	400/350/350
Forward Voltage	3.4V/2.5V	Viewing Angle	120°
LED Package	14.5 mm x 7.5 mm	Lumens	55/60/20
Maximum Forward Voltage	2.8V/3.8V/3.8V	Maximum LED Junction Temperature	120°C
Millicandela	43000mcd	Operating Temperature(°C)	-35~+50
Standard Rated Current	350mA	Reverse Voltage	5V
Storage Temperature(°C)	-35~+50	Power Output	3W (1W per channel)
Wavelength	625nm/525nm/465nm	Peak Forward Current	800mA



- Voltage: RED 2.2-2.7, GREEN 3.2-3.8, BLUE 3.2-3.8
- Color: 620-630nm 515-525, 465-475 (nm)
- Lumens: R 35, G 45, B 12 Average (lm)
- Angle: 110-130 Deg
- 750mA Forward Current

Lambertian type: $\Phi 8.0 \times 5.4 \text{ mm}$

- Viewing angle: 120°
- Power: 3W
- Emitting color: full color (red, green, blue)
- Mainly used in lighting area

Long operating life (up to 50,000 hours, depend on operation condition)
More energy efficient than incandescent and most halogen lamps
Low forward voltage operated
Instant light (less than 100ns)
No UV
High ESD protection >4000V
IR reflow soldering
Lead (pb) free and RoHS compliance
Typical Applications
Reading lights
Portable flashlight
Architecture lighting and room lighting
Illuminated advertising
Emergency light
Decorative and entertainment lighting
Uplighters and Downlighters
Bollards / Security / Garden lighting
Indoor and Outdoor commercial lighting
LCD backlights / Light guides/Automotive
General lighting (Street lamp / Tunnel light / Floodlight, ect.)

ANEXO – B



Technical Specification

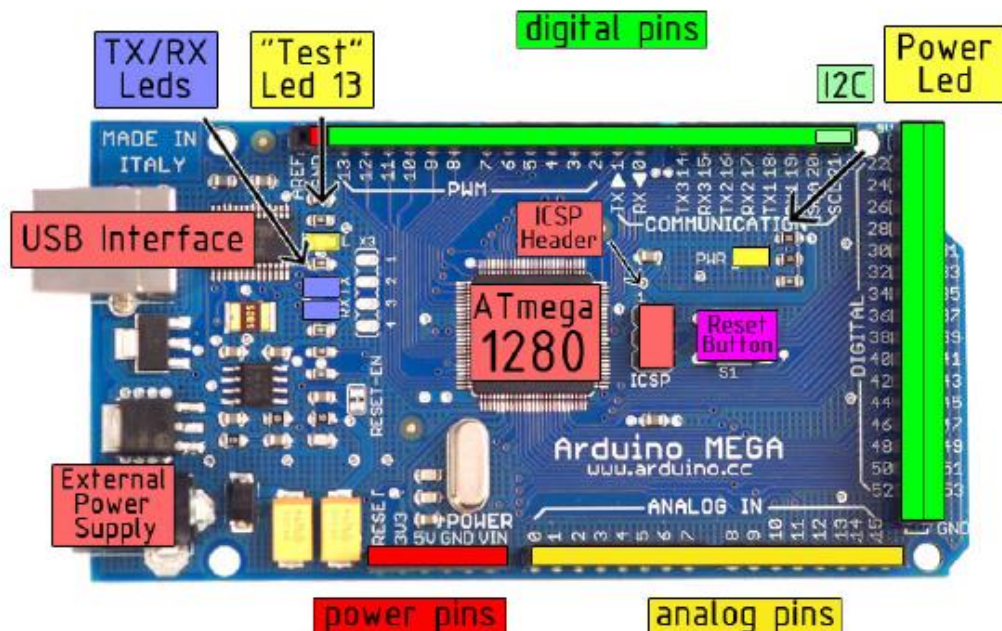


EAGLE files: [arduino-mega-reference-design.zip](#) Schematic: [arduino-mega-schematic.pdf](#)

Summary

ATmega1280	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega1280 datasheet.

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega1280 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable.**

Blink led

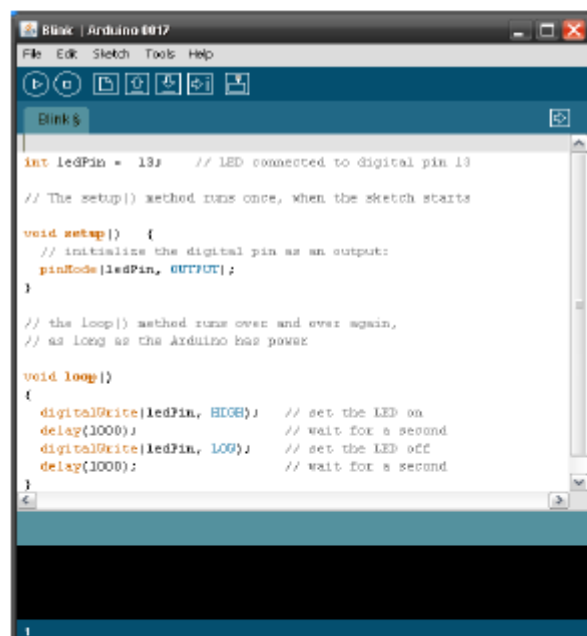
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

Dimensioned Drawing

